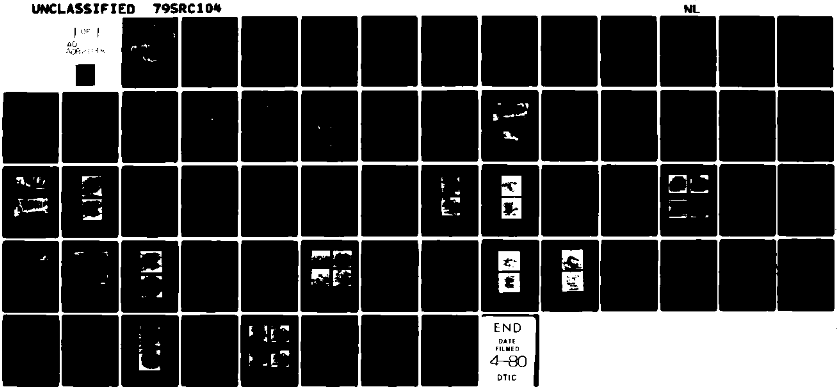
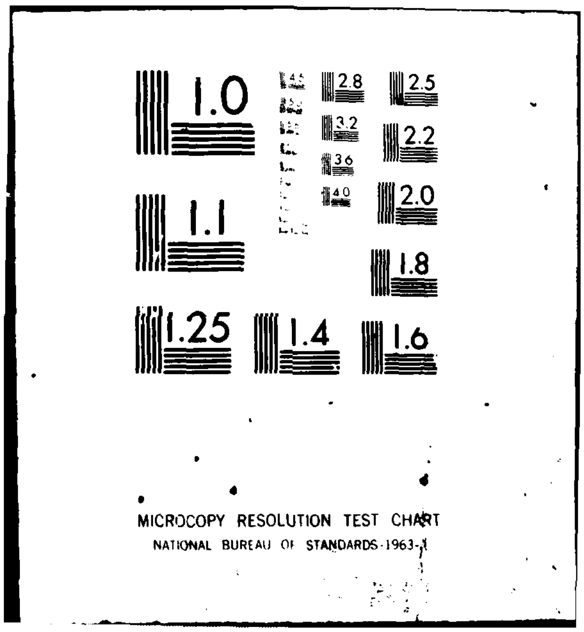
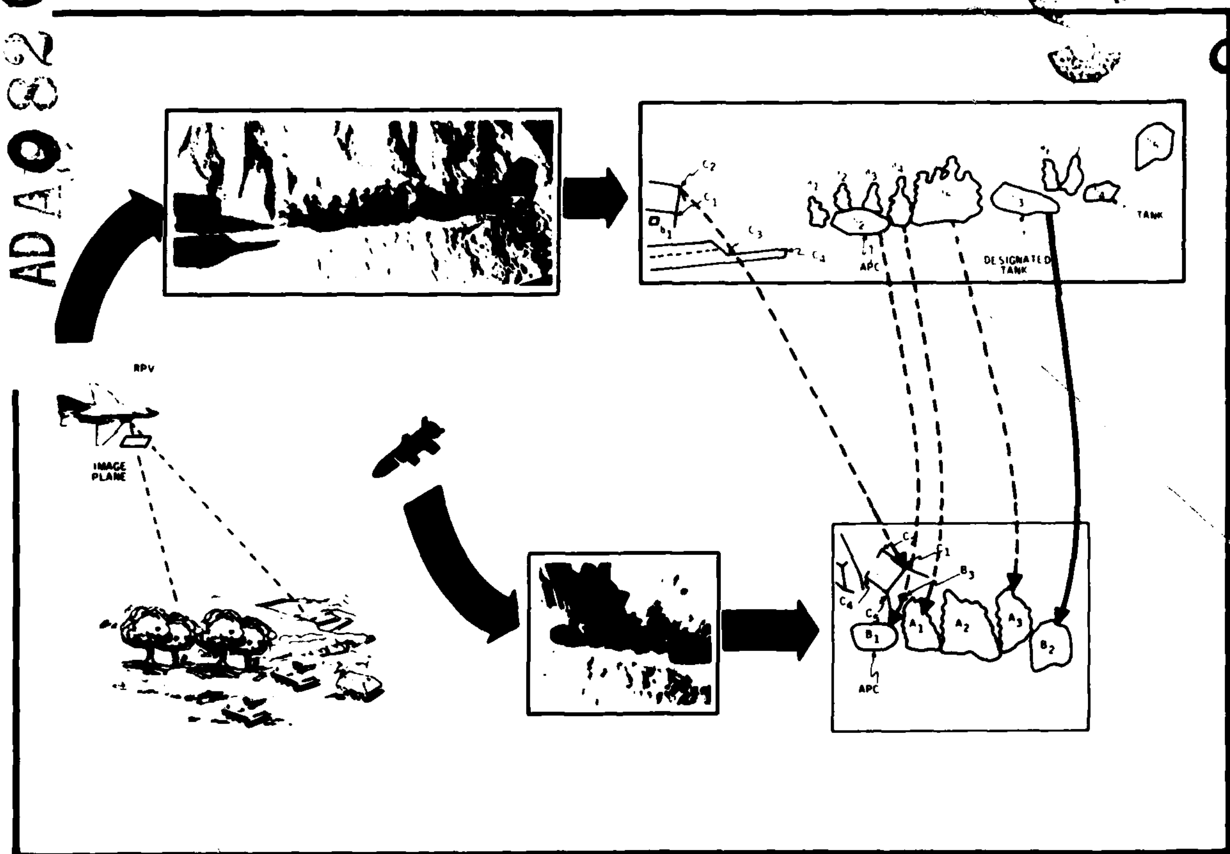
AD-A082 038 HONEYWELL SYSTEMS AND RESEARCH CENTER MINNEAPOLIS MN F/G 17/7
ADVANCED PATTERN MATCHING TECHNIQUES FOR AUTONOMOUS ACQUISITION--ETC(U)
DEC 79 P M NARENDRA, J J GRABAU DAAK70-79-C-0114
UNCLASSIFIED 79SRC104 NL




1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ADVANCED PATTERN MATCHING TECHNIQUES FOR AUTONOMOUS ACQUISITION

First Quarterly Progress Report



by

P.M. Narendra
J.J. Grabau

Honeywell

SYSTEMS & RESEARCH CENTER
2600 RIDGWAY PARKWAY
MINNEAPOLIS, MINNESOTA 55413

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	9
4. TITLE (and Subtitle)	6 ADVANCED PATTERN MATCHING TECHNIQUES FOR AUTONOMOUS ACQUISITION.
5. TYPE OF REPORT & PERIOD COVERED	Quarterly Progress Report, No. 1 23 Aug ~~to~~ 23 Nov [redacted]79
6. PERFORMING ORG. REPORT NUMBER	14 79SRC104
7. AUTHOR(s)	10 P. M. Narendra, J. J. Grabau
8. CONTRACT OR GRANT NUMBER(s)	15 DAAK70-79-C-0114
9. PERFORMING ORGANIZATION NAME AND ADDRESS	Honeywell Inc., Systems and Research Center, 2600 Ridgway Parkway, Minneapolis, Minnesota 55413
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	12 63
11. CONTROLLING OFFICE NAME AND ADDRESS	Night Vision and Electro-Optics Laboratory, Fort Belvoir, Virginia 22060
12. REPORT DATE	11 Dec~~ember 19~~79
13. NUMBER OF PAGES	61
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	
15. SECURITY CLASS. (of this report)	Unclassified
15a. DECLASSIFICATION DOWNGRADING SCHEDULE	

16. DISTRIBUTION STATEMENT (of this Report)

Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Mr. C. Goehrig is NV&EOL Contract Monitor on this program.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Infrared	Target recognition	Symbolic processing
FLIR	Pattern recognition	Target tracking
Target cueing	Image processing	Scene analysis
Target screening	Artificial intelligence	Pattern matching

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This is the first interim quarterly progress report on "Advanced Pattern Matching Concepts," NV&EOL contract No. DAAK70-79-C-0114. It reports the results of work performed between August 23, 1979 and November 23, 1979.

(continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

JOB

402349

20. ABSTRACT (continued)

The key objective of this effort is the development of pattern-matching algorithms which can impart autonomous acquisition capability to precision-guided munitions such as Copperhead and Hellfire. Autonomous acquisition through pattern-matching holds the promise of eliminating laser designation and enhancing fire power by multiple target prioritization.

The pattern-matching approach being developed under this program is based on a symbolic pattern-matching framework, which is suited for the autonomous acquisition scenario. It is based on matching a symbolic representation derived from the two images, and it can accommodate the stringent pattern-matching criteria established by the scenario: enormous differences in the scene perspective, aspect and range between the two sensors, differences in sensor characteristics and illumination, and scene changes such as target motion and obscuration from one view point to the other.

CONTENTS

CONTENTS (concluded)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (concluded)

SECTION I

INTRODUCTION

This is the first interim quarterly progress report on "Advanced Pattern Matching Concepts," NV&EOL contract No. DAAK70-79-C-0114. It reports the results of work performed between August 23, 1979 and November 23, 1979.

The key objective of this effort is the development of pattern-matching algorithms which can impart autonomous acquisition capability to precision-guided munitions such as Copperhead and Hellfire. Autonomous acquisition through pattern matching holds the promise of eliminating laser designation and enhancing fire power by multiple-target prioritization. However, this application imposes stringent performance requirements on the pattern-matching algorithm--it must be robust under perspective and aspect change, target motion, illumination change, sensor differences, image quality, and target obscuration. Conventional pattern-matching techniques are incapable of meeting the performance requirements in the autonomous acquisition scenario.

The pattern-matching approach being developed under this program is based on a symbolic pattern-matching framework which is suited for the autonomous acquisition scenario. It is based on matching a symbolic representation derived from the two images rather than on numerical correlation. It can accommodate the stringent pattern-matching criteria established by the scenario: enormous differences in the scene perspective,

aspect and range between the two sensors, differences in sensor characteristics and illumination, and scene changes such as target motion and obscuration from one view point to the other.

Figure 1 shows a broad overview of the symbolic pattern-matching approach. The symbolic pattern-matching technique is based on matching a symbolic representation of the two images, not the gray levels of the individual picture elements themselves, as in conventional correlation approaches. A symbolic representation of an image consists of describing objects (or distinctive elements) in the image and their positions.

The symbolic matching technique operates on the symbolic image to find an optimal match which simultaneously ensures 1) that the matched objects are similar in their descriptors and 2) that the interobject configuration

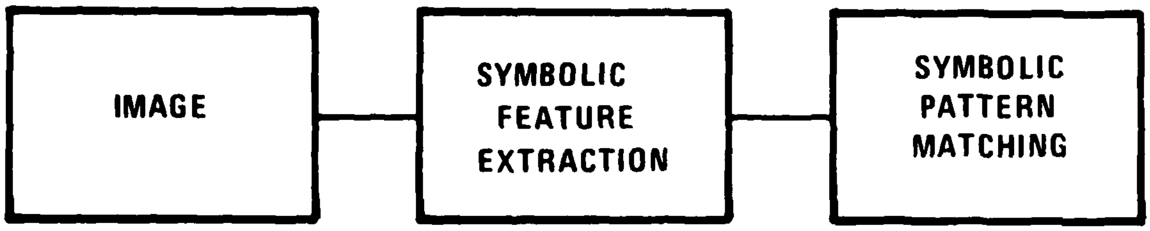


Figure 1. Pattern-Matching Overview

in the two matched sets of objects are consistent. This yields a robust pattern-matching algorithm which is insensitive to a number of variables, including variation in object descriptors.

SUMMARY OF PROGRESS

The following results have been accomplished in this reporting period:

1. A data base of 16 photographs of a terrain model characterizing various scene conditions has been digitized. An additional FLIR video sequence of tactical targets in natural terrain has also been digitized to serve as input to the pattern-matching simulation task.

2. Symbolic object feature extraction software is complete. This includes features derived from the segments extracted from two segmentation schemes, Prototype Automatic Target Screener (PATS) and Prototype Similarity. Software developed under previous Honeywell contracts and IR&D efforts was extended to measure the object features needed in the matching scheme.

3. Edge-based feature extraction software is complete. New software has been generated for Sobel edge extraction, maximum directional matched filtering of the thresholded edges, connecting collinear edge points, and merging shorter edge segments into long connected lines. This software has been evaluated on test images and the results are included.

4. Work has begun on symbolic object-matching criteria and the branch and bound algorithm for matching object configurations.

REPORT OUTLINE

The body of the report is organized under the following headings:

- Overview of approach
- Pattern-matching data base
- Symbolic object descriptors
- Plans for the next reporting period

SECTION II

OVERVIEW OF APPROACH

This section briefly reviews the two driving applications for the pattern-matching algorithms under development--the Copperhead and Hellfire scenarios. This review serves to set the basic performance requirements for the pattern-matching algorithms and shows how pattern-matching can be used for semiautonomous acquisition. This review is followed by an overview of the program approach which places the reported results in the following sections in the proper context.

Precision-guided munitions such as Copperhead and Hellfire currently use laser designation of the targets for terminal homing. Here, a remote designator illuminates the target by a laser, which is then sought by the seeker module on the projectile/missile as it enters the target basket. The seeker gyro then generates the guidance controls which are used to maneuver the target to a direct hit.

Laser designation of a target has a number of practical deficiencies which make autonomous acquisition of the targets highly desirable. Some of the drawbacks of laser designation are as follows:

- The remote designator (infantryman, scout helicopter, AH64 attack helicopter, or the RPV) has to hold the laser on target from the time of initial reconnaissance to the time of impact. Therefore, the designator is exposed to enemy fire for a significant length of time.

- Simultaneous engagement of multiple targets requires multiple remote designators. This is again because the laser designation has to remain on the target for the entire duration of designator/ launcher handshake and projectile launch until impact. This may result in under-utilization of the artillery capability.
- It is often difficult to hold the laser spot on the target, resulting in spot jitter and, therefore, a low probability of direct hit.

Using autonomous acquisition, we can dispense with laser designation. However, to achieve automatic acquisition, it is not sufficient just to be able to recognize a class of targets in the projectile's field of view (FOV). As noted above, multiple target engagement requires the projectile to engage a specific target in the basket. This would allow successive projectiles to be programmed to hit different targets, enhancing the effectiveness of the fire power. Moreover, the imaging sensor in the projectile typically has a small FOV (limited by the size of the optics). Therefore, when the projectile enters the basket, the indicated target may not even be in its FOV. If this happens, target recognition capability alone does not suffice to guide the projectile to the target. The projectile may have to be guided so that it does acquire the designated target in its FOV before it can be guided to its target.

The program is developing the advanced pattern-matching techniques to achieve this autonomous acquisition capability needed for multiple target scenarios. Both the Copperhead and Hellfire have similar requirements for autonomous acquisition in that these systems currently require a remote designator to illuminate the target for the munition's laser seeker.

AUTONOMOUS ACQUISITION SCENARIO

The following scenario addresses pattern-matching for autonomous acquisition in the Copperhead projectile. A remotely piloted vehicle (RPV) loitering over the battlefield area observes the target basket with an imaging sensor (FLIR/TV) and transmits its image back to a ground station. The ground station tracks the RPV position with a line-of-sight tracker. This ground station can be several kilometers behind the forward edge of the battlefield (FEBA). This step is depicted by the first picture in the sequence in Figure 2a to 2d.

At the ground station, the received image is reconstructed, enhanced, and processed through an automatic cuer to detect potential targets in the FOV. This cued image is displayed in real time on a video monitor to an operator.

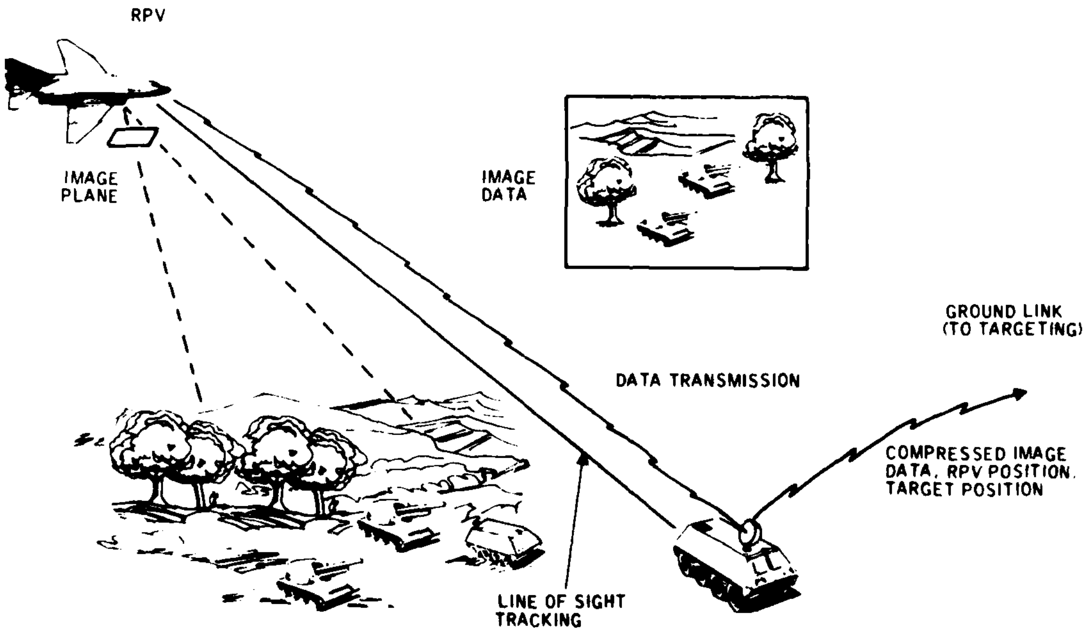


Figure 2a. RPV Images Basket

The operator indicates on the display a specific target which the projectile is intended to hit. The image is then further processed to a symbolic representation of the RPV's FOV which includes information such as a parameteric representation of all the objects in the FOV and their positions. Information on the approximate position and direction of viewing of the RPV and other world knowledge is also incorporated into this representation. This symbolic representation collectively forms a template, no longer in an image format, which is forwarded to an artillery gunner as shown in Figure 2b. The template is then programmed into the electronics of the projectile's seeker, and the projectile is fired into the target basket based on the target coordinates supplied by the RPV. This is shown in Figure 2c.

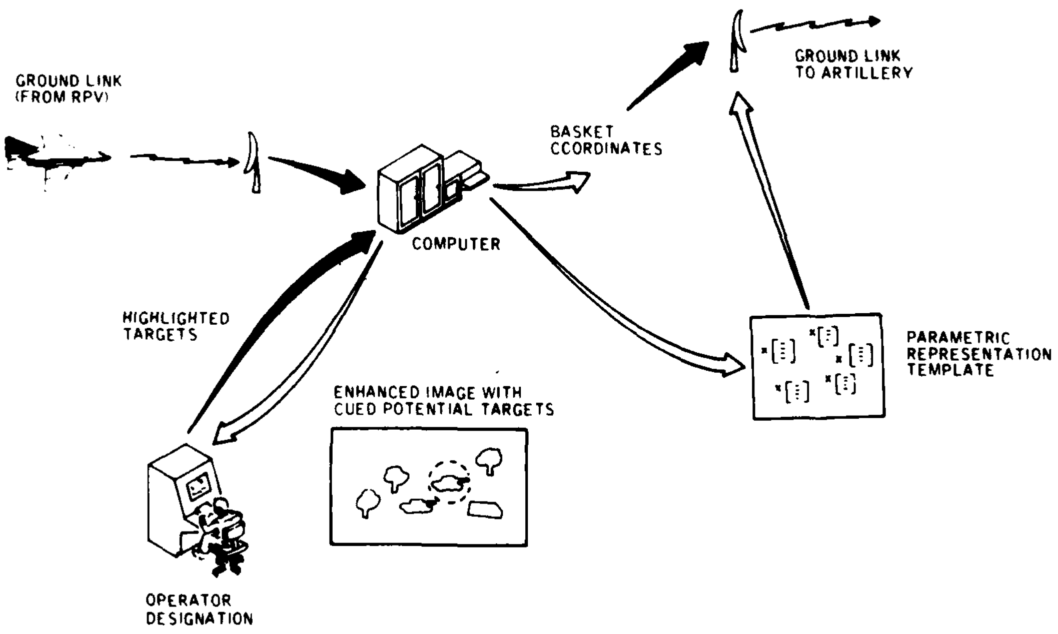


Figure 2b. Target Designation and Ground Processing For Reference Template

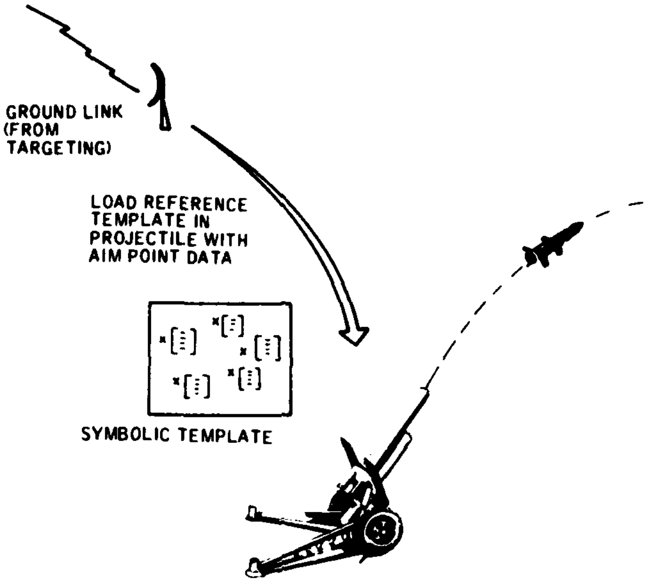


Figure 2c. Load Template and Fire Projectile

As the projectile approaches the basket, its sensor is activated and a portion of the basket becomes visible in its FOV. The symbolic pattern-matching algorithm being developed is then used to match the configuration of objects found in the projectile's FOV to a subset of the symbolic reference template from the RPV's image (Figure 2d). This orients the projectile's FOV with respect to the reference template and finds the designated target in the projectile's FOV.

The projectile's sensor will have a smaller FOV than that of the RPV, and therefore, the indicated target may not be in the current FOV of the projectile. Here, the position of the projectile's FOV found by matching may be used to maneuver the projectile so that the target does appear in the projectile's FOV in successive frames. Further, if the indicated target is obscured (or not extracted by the segmenter in the projectile) the configuration of other objects can be matched with the reference template to find

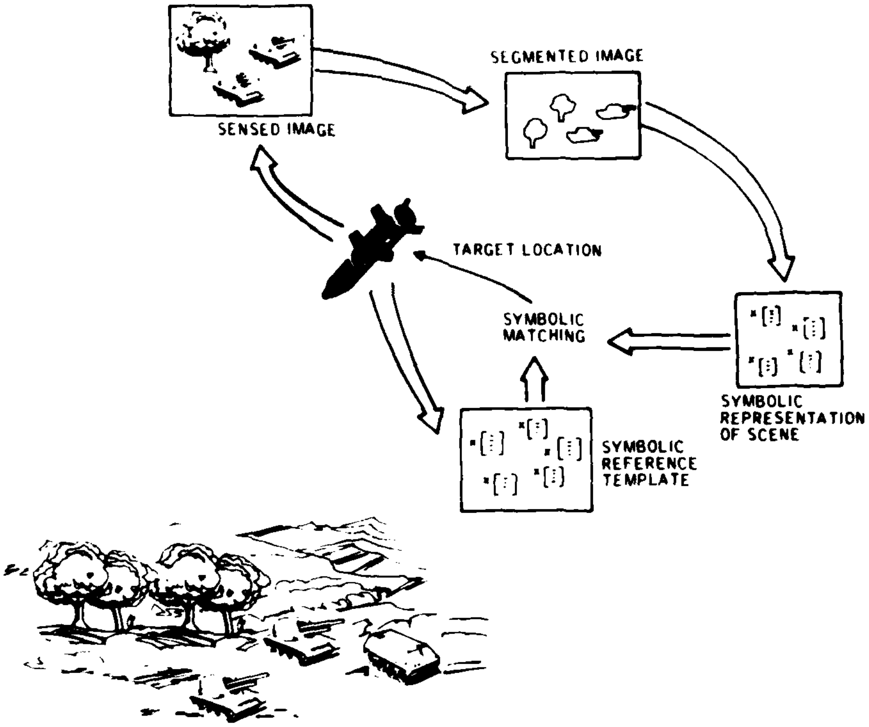


Figure 2d. *Pattern Matching* in Projectile for Autonomous Acquisition

the location of the indicated target in the projectile's FOV. This is then used to guide the projectile to impact. This is important because the designated target may be obscured by smoke, rain, trees, and sensor noise which may prevent its extraction. This makes it essential to be able to match configurations of objects rather than to find the designated target alone.

The pattern-matching concept for autonomous acquisition is equally valid in the Hellfire missile scenario as well. Figure 3 shows a typical Hellfire scenario, where the Attack Helicopter (an AH64) pops up, takes a "snapshot" of the target basket, dives under cover, programs the reference template of the basket into the Hellfire missile, and launches it. Here again, matching the missile's FOV with the reference template guides it to target engagement.

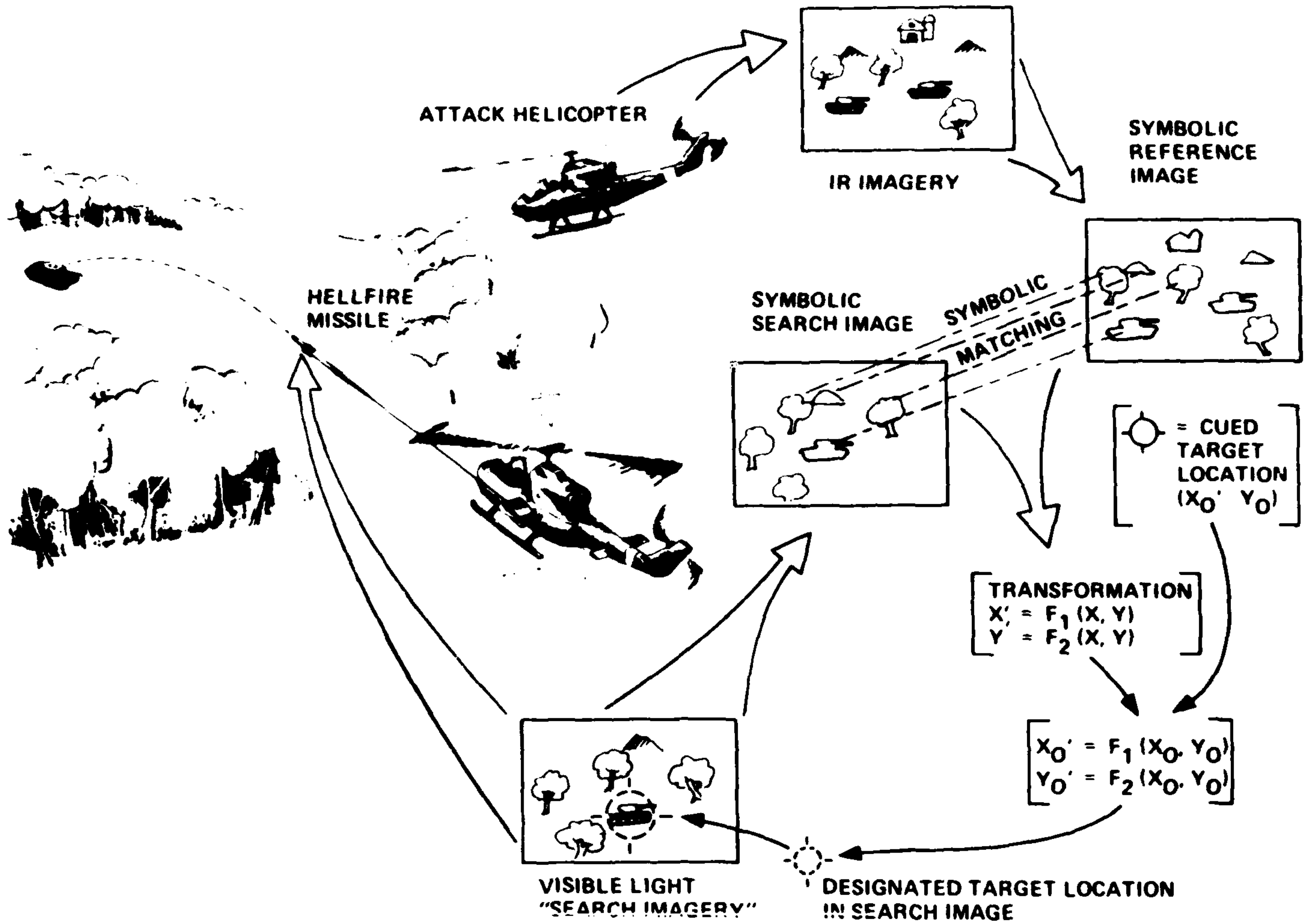


Figure 3. The Hellfire Autonomous Acquisition Scenario With Symbolic Pattern Matching

We see that through the help of pattern matching, we can eliminate the problems associated with laser designation in the Copperhead and Hellfire scenarios. First, the need to hold a laser spot on the target is eliminated. Once the RPV (or other remote) image has been transmitted, there is no need to expose the RPV. Second, and perhaps more important, multiple target engagement becomes a reality. Targets can be prioritized and separately designated on successive projectiles so that there will be no duplication of hits. Multiple targets can be engaged as fast as we can produce templates. Further, one RPV can service many artillery crews because the RPV can keep moving and looking. There is no need to wait for one target to be destroyed before looking for another one.

This scenario highlights several operational requirements for the pattern-matching algorithm. They include successful matching under

- large differences in imaging sensor geometries--perspective, aspect, range, and sensor roll angle.
- widely differing sensor characteristics--wavelength (IR/visible), resolution, and FOV.
- illumination variations (for visible sensors).
- target obscuration and scene change (due to staleness).

The symbolic-matching approach has the potential of meeting these requirements. The following is an overview of our general approach, which puts the results in subsequent sections in proper context.

OVERVIEW OF PATTERN MATCHING

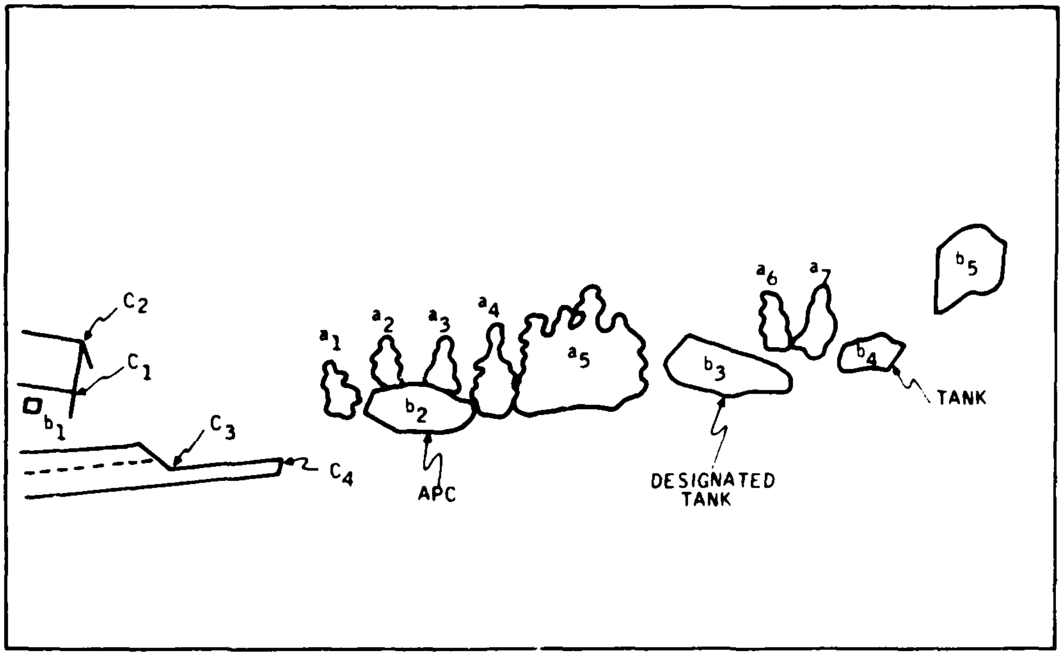
Figure 4 shows the basic steps in symbolic pattern matching between the two images representing the RPV and projectile FOV. The first step is to extract object features from both images. Note that the result of this scene analysis combines both blobs (from a target screener segmenter) and edge-based features such as long lines and vertices as well. The next stage is the symbolic description of these objects (blobs and lines) as lists of object descriptors and their positions in the two FOV. Finally, symbolic pattern matching is performed between the two symbolic images to establish correspondence between the target designated on the RPV image and the corresponding object in the projectile's image. The key to a robust symbolic matching is that in an optimal match both the object descriptions and the inter-object relationships in the two images should be consistent. Criteria for evaluating interobject configurations matches must be able to account for scene perspective, aspect, roll, and scale differences. The development of an efficient search algorithm to examine the most promising object and configuration matches as defined by the match criteria is the main thrust of this program.

The subsequent sections present the progress made in the first quarter toward the above program objective. This includes data base generation, object segment and edge feature extraction softw^re, and data structures for symbolic descriptions of these features.

RPV Image

Projectile's Image

Figure 4a. An Overview of Advanced Pattern-Matching Approach For Semi-Autonomous Acquisition



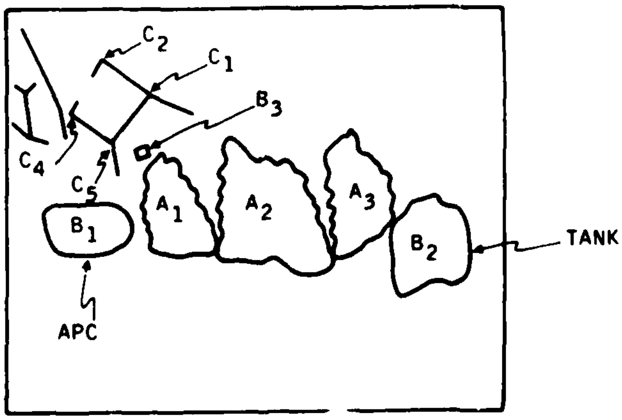


Figure 4b. Feature Extraction (Edge and Segmented Objects)

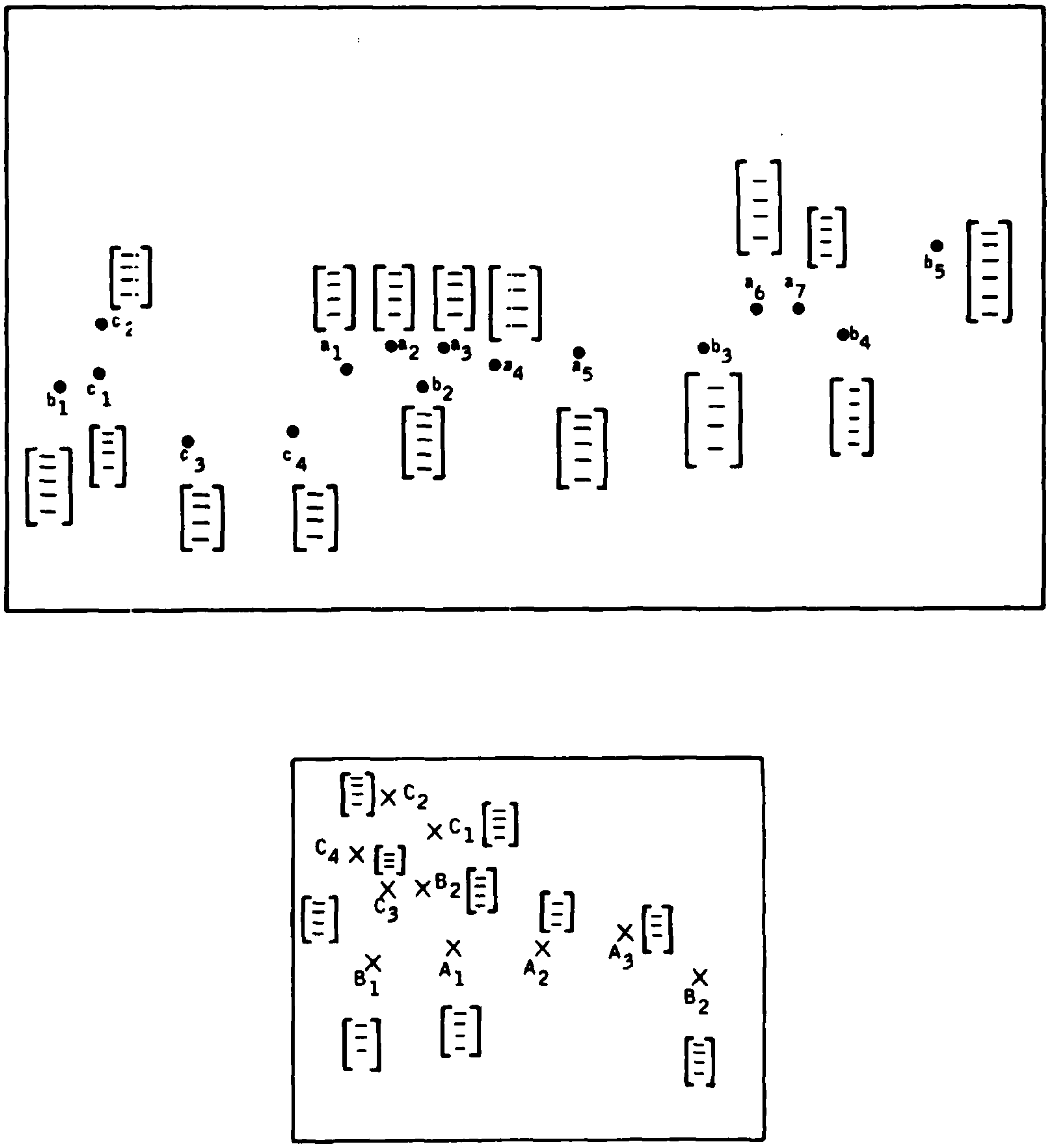


Figure 4c. Symbolic Description of Images

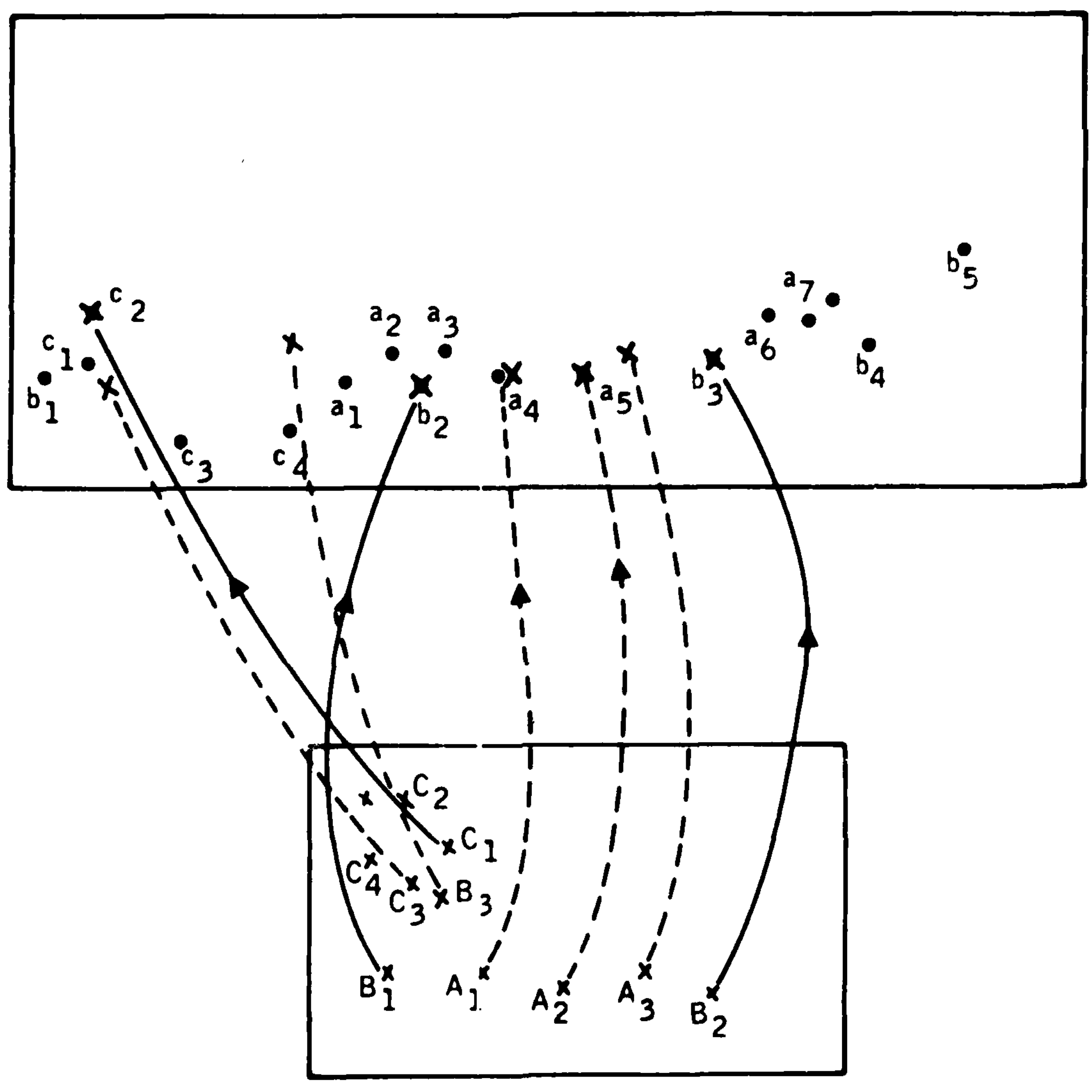


Figure 4d. Symbolic Pattern Matching

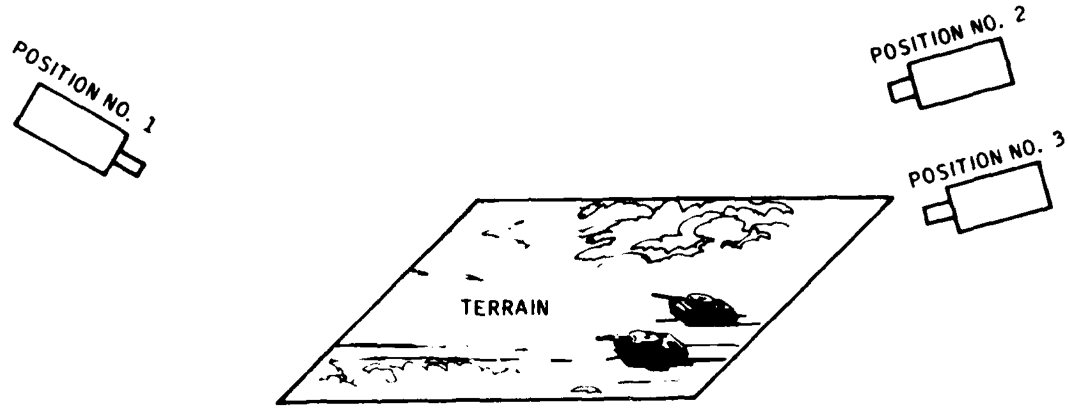
SECTION III

PATTERN-MATCHING DATA BASE

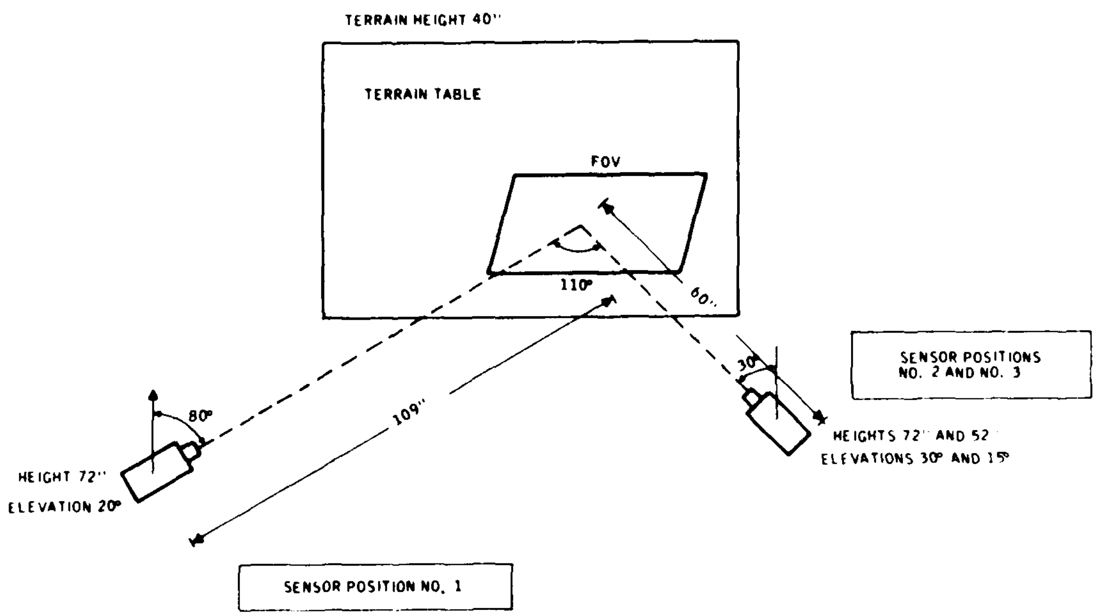
As we saw in the operational requirements summarized in Section II, the pattern-matching algorithms must be capable of successfully matching images from a variety of sensor geometries and sensor types. For the simulation task, ideally, we need images of the same scene from FLIR and TV sensors, representing perspective changes from 0-180 deg, aspect changes of 1 to 90 deg, and a variety of illumination conditions. This section reports the status of a preliminary pattern-matching data base which has been gathered under this task.

Two sources of data have been exploited so far. The first is a collection of photographs of a terrain table from two viewpoints 110 deg apart and a variety of illumination conditions (Figure 5). Sixteen of these photographs have been digitized and have been used as test images in the object feature extraction effort. Figure 6 is a sample from this digitized data base.

The second source of pattern-matching imagery is a 525 line FLIR video tape supplied by NV&EOL. It consists of a LOHTADS helicopter approach over a set of stationary targets (including tanks and APCs). Two digitized frames from this tape are shown in Figure 7. Because the sensor angle varies as the helicopter goes by the targets, these images represent a significant charge in aspect and angles, as well as range, and will serve as a preliminary pattern-matching FLIR data base.

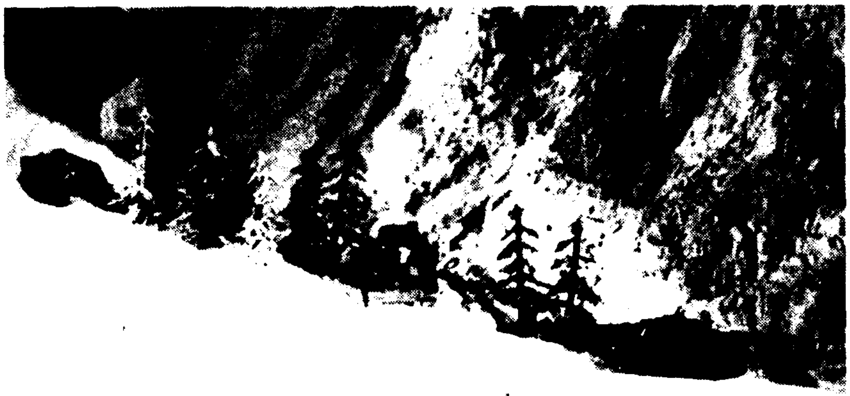


TERRAIN BOARD LAYOUT



CAMERA STATIONS

Figure 5. Geometry for Pattern-Matching Photograph Data Base



a.

b.

Figure 6. Examples of Pattern Matching Photographs Data Base (Terrain table)

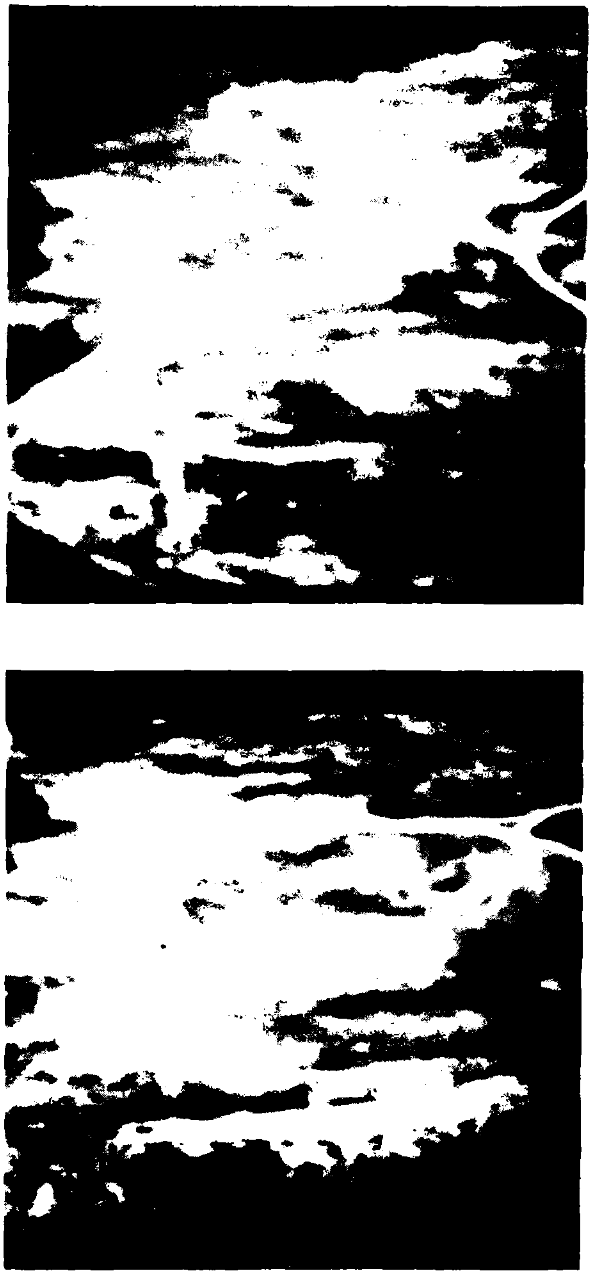


Figure 7. Two Digitized Frames From a FLIR Video Tape of a Helicopter Approach Over a Set of Stationary Targets

As we will see in the next section, FLIR images are far more easily segmentable than visible imagery. Since the major emphasis of this program is pattern matching, we feel that FLIR imagery should allow us to concentrate more on pattern matching per se. To this end, we are searching our FLIR video tape library for candidate pattern-matching sequences. The PATS training data collection being undertaken by NV&EOL should also provide FLIR imagery of target formations from different aspects and perspectives. We will attempt to digitize this 875 line FLIR video using the PATS digitizer in the next reporting period.

SECTION IV

SYMBOLIC OBJECT DESCRIPTORS

The first step in symbolic matching is the extraction of distinctive scene components or objects by an automated segmentation procedure from the RPV image and the projectile image as shown in Figure 4 in the overview section. As noted in Figure 4, the result of the scene analysis can be either blobs from a segmentation algorithm or edge-based linear features such as straight lines and vertices (Figure 8). These object representations in the reference template and the projectile image form the starting point for matching objects in the two images. As the first step, based on the object descriptors alone, every object in the projectile image is compared with the objects in the reference template to establish a list of "best matches" for each object. In performing this comparison, the representation of the objects in the template are compared with the object descriptors extracted in the projectile's image. Object matching at the local level achieves a drastic reduction in the number of object configurations that have to be tested for interobject consistency. Object descriptors are also used in evaluating configuration matches as well.

This section reports the results of the object feature extraction task. In this reporting period, software has been completed and evaluated for the extraction of object descriptors from both segmentation schemes (blobs) and edge-based line features. In particular, the Prototype Similarity[1]

[1] Developed under Automated Imagery Recognition System, DARPA Contract No. F33615-76-C-1324.

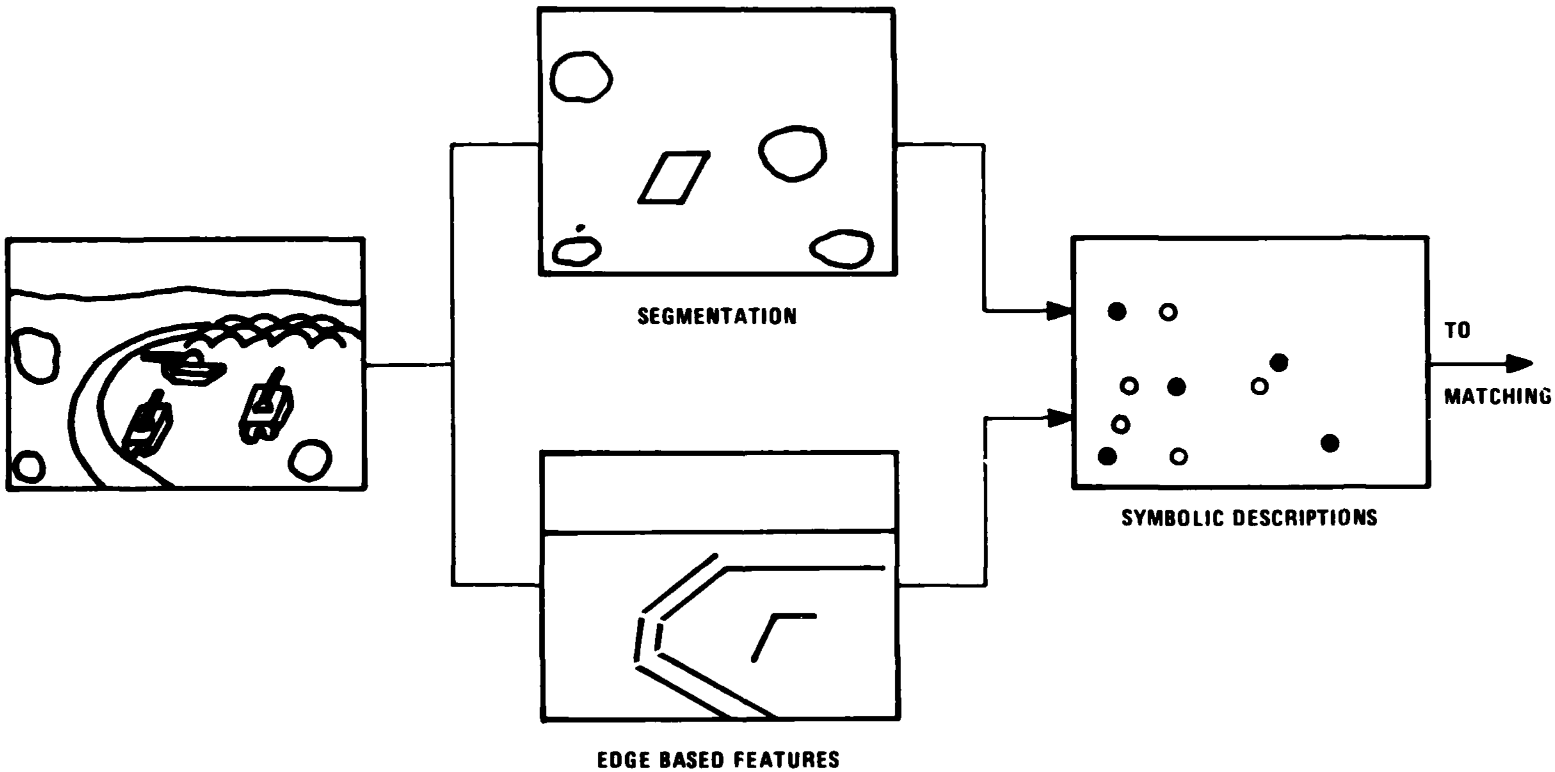


Figure 8. Symbolic Feature Extraction for Pattern Matching

segmentation and the PATS[2] segmentation software have been extended to measure pattern-matching object descriptors. New software has been developed specifically under this program for edge-based feature extraction --Sobel Edge Operator, maximum directional matched filter, edge-linking algorithm, and line-merging algorithm. This software is now being evaluated on the test imagery, and the results are included in the discussion which follows.

We are pursuing the use of two types of features in the symbolic pattern-matching algorithm--segmented object features and edge-based features. Extraction of object features will be described next, followed by a discussion of the extraction of edge-based features.

SEGMENTATION ALGORITHMS FOR OBJECT FEATURE EXTRACTION

Two segmentation algorithms are candidates for use in pattern matching under this program. They are Prototype Similarity and PATS. PATS is an acronym for Prototype Automatic Target Screener.

Prototype Similarity

The basic idea of Prototype Similarity is to obtain a set of prototypical regions in the image that are in a sense maximally dissimilar and provide therefore, a parsimonious explanation of the differences between cells in the image. It was originally developed for FLIR images. A priori

---

[2] Developed under Prototype Automatic Target Screener (PATS) Contract No. DAAG53-77-Q-1252.

information about the image is used to select the first two prototypes. In the case of FLIR data for images of tactical interest, the *a priori* information is that the hottest spots in the image are probably running motors in targets, whereas the coolest spots are probably background. After using *a priori* information to select the first two prototypes, succeeding prototypes are selected by a process that tends to find prototypes not similar to those already found. This process continues until no more prototypes can be generated. The *a priori* information is then used to infer meanings for the prototypes. This process can be repeated for more than one feature. For example, prototypes can be obtained for image intensity and for image gradient.

The meanings inferred for the cells in the intensity image and in the gradient image can then be combined to give a composite interpretation of the meanings of the cells. Typically, the meanings inferred are target, background, and edge.[3] The version of Prototype Similarity being used in this project next proceeds to connect adjacent cells with the same meaning into regions, generate an image with the different regions labeled, and output a list of the regions. The labeled image and the list are inputs to feature extraction software developed under the present program. The labeled images generated by Prototype Similarity from the images in Figure 9 are shown in Figure 10.

---

[3] R.K. Aggarwal, "Adaptive Image Segmentation Using Prototype Similarity," Proceedings of the Pattern Recognition and Image Processing Conference, Chicago, Ill. May 31 to June 2, 1978. pp. 354-359.

Figure 9. Original Test Images for Prototype Similarity Object Feature Extraction and for Edge Feature Extraction

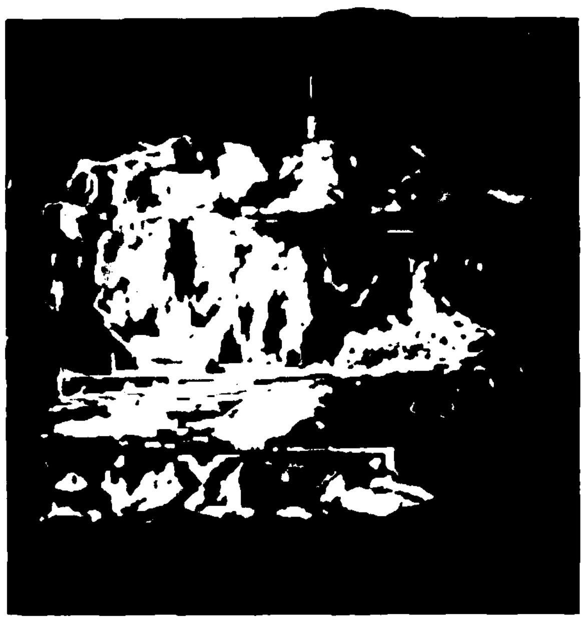


Figure 10. Labeled Images Generated by Prototype Similarity for the Images in Figure 9

An object feature extraction program which operates on the output of the segmentation program has been completed. The feature extraction program reads the list generated by Prototype Similarity and makes a decision for each region as to whether to process the region to extract features. This decision is based on information in the list about the meaning of the region, the area of the region, and the location of the bounding rectangle containing the region. Background regions are not usually processed; either excessively large or excessively small regions can be left unprocessed.

The feature extraction process for an accepted region proceeds in two steps. In the first step the border of the region is traced to determine the perimeter of the region.

In the second step, features that depend on all the pixels of the region are obtained. These features include the following:

1. Average intensity
2. Variance of intensity
3. Centroid of the region
4. Second moments of the silhouette of the region

In addition, the following features are provided by Prototype Similarity:

1. Area of the region

2. Inferred meaning of the region

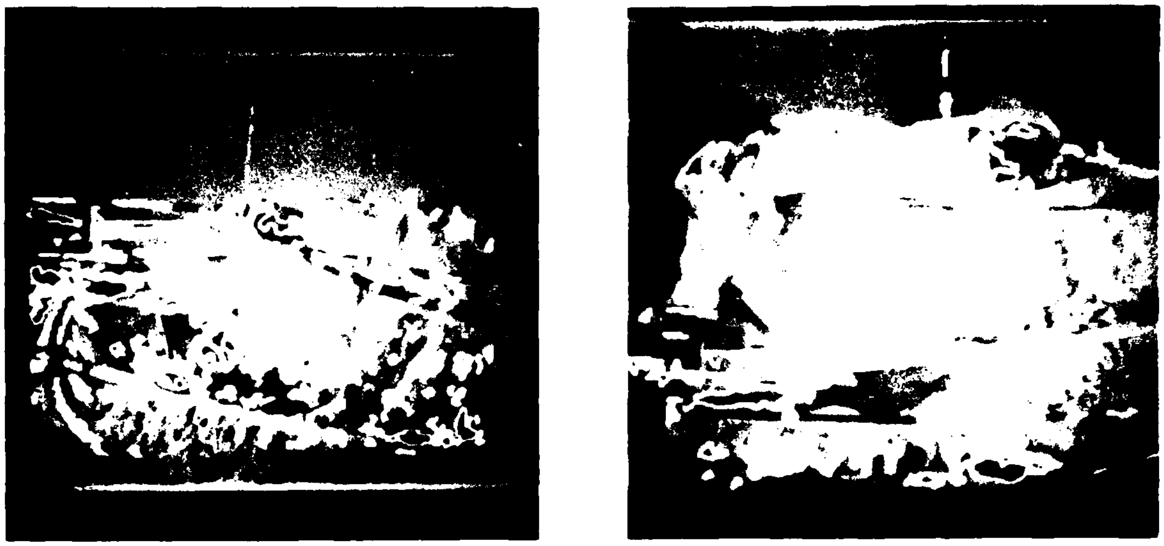
3. Minimal bounding rectangle for the region which gives data such as the length and width of the region

The centroid data will be used by the matching algorithm for determining the matching transformation. The other data will serve to evaluate the likelihood that pairs of regions from the two images are matching regions. Outlines and centroids of regions processed by the feature extractor for the segmentations of Figure 10 are shown in Figure 11.

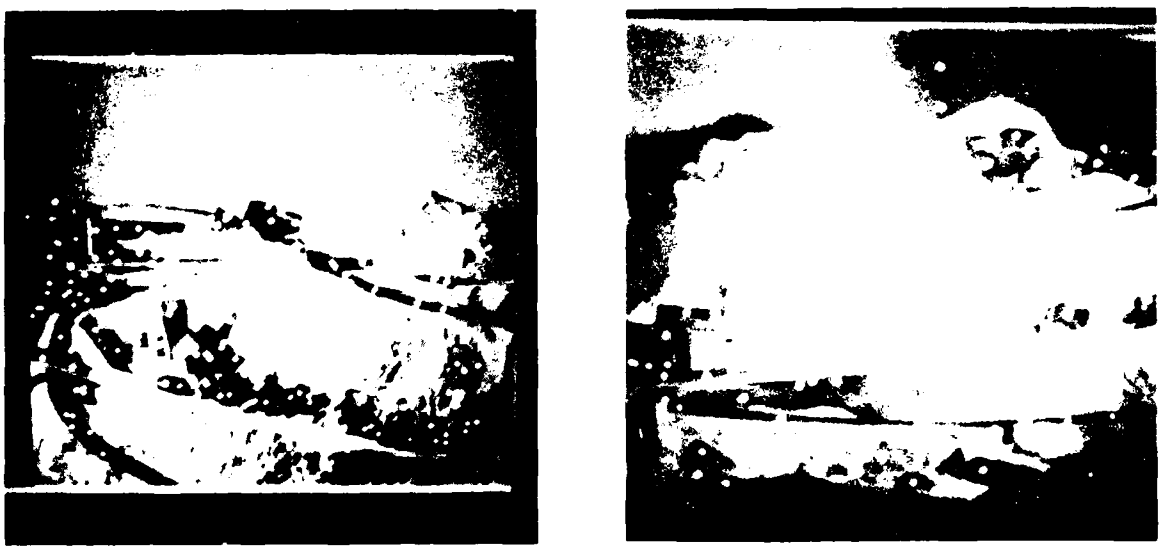
PATS Segmentation

The PATS segmentation scheme uses information about the hot and cold areas and the edges in an image to extract objects distinct from the background. It was developed for FLIR images.

Significant parts of the background in FLIR images are often nearly as hot as the hot targets to be extracted or approximately as cold as the cold targets to be extracted. This makes it impossible to extract hot and cold targets using only one global set of thresholds applied directly to the original image. PATS uses a recursive filter to estimate the background intensity adaptively. It then thresholds the deviation of the image intensity from the estimated background intensity to extract hot and cold regions. This results in two binary images--one a map of the hot regions and the other a map of the cold regions. PATS also produces a binary map of the



a. Outlines



b. Centroids

Figure 11. Outlines and Centroids of the Regions of the Images in Figure 10 That Were Processed by the Object Feature Extraction Algorithms

edge points in the image. By comparing the hot and cold maps with the edge map and requiring that extracted hot and cold objects have their boundaries be delineated by edge points, PATS reduces the clutter in the output image and also does a better job of determining the shapes of the extracted regions.

The results of applying PATS to a FLIR image are shown in Figure 12. PATS software has been converted to run on the Honeywell Level 6 mini-computer based image processing facility under the Advanced Target Tracker Concepts program.[4] The converted version of PATS extracts all the features presently being extracted for Prototype Similarity plus other features such as Fourier descriptors, higher order moments of the intensity and the silhouette, and also moments of the object boundary.

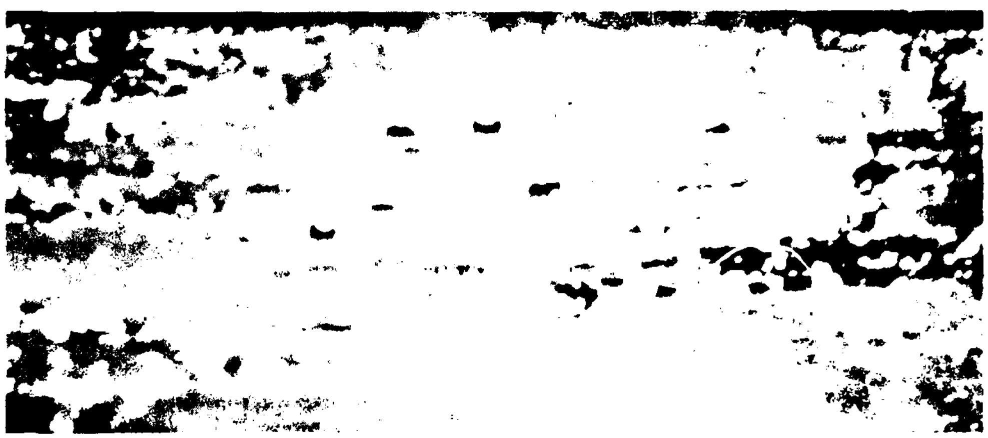
EDGE-BASED FEATURES

There are good reasons to believe that the pattern-matching algorithm should be able to exploit edge features--especially long edges. Many natural scenes contain long edges because of the presence of roads, rivers, lakes, seashores, tree lines, and other extensive regions. One such scene is shown in Figure 13, together with the Sobel edge image for that scene. Roads and rivers are particularly likely to appear in scenes of tactical interest. In addition to being likely to appear in scenes of interest, long lines are powerful cues for matching because of the considerable information

---

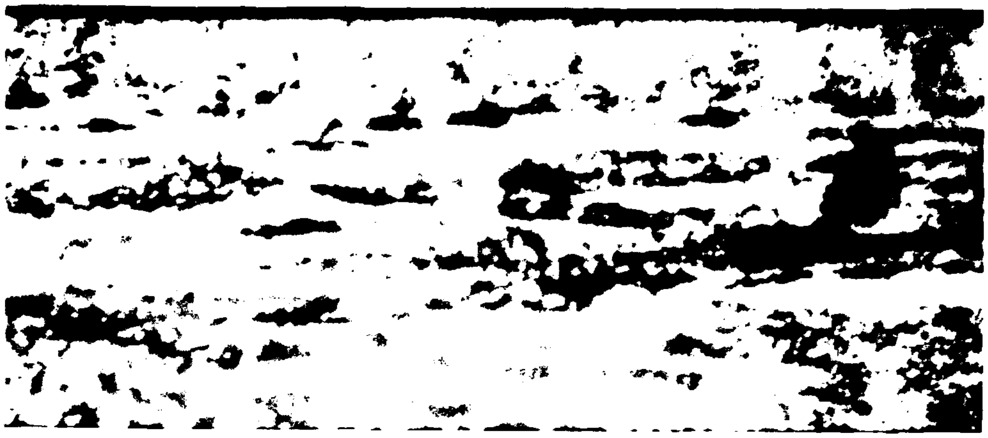
[4] NV&EOL Contract Number DAAK70-79-C-0150

a. Original Image

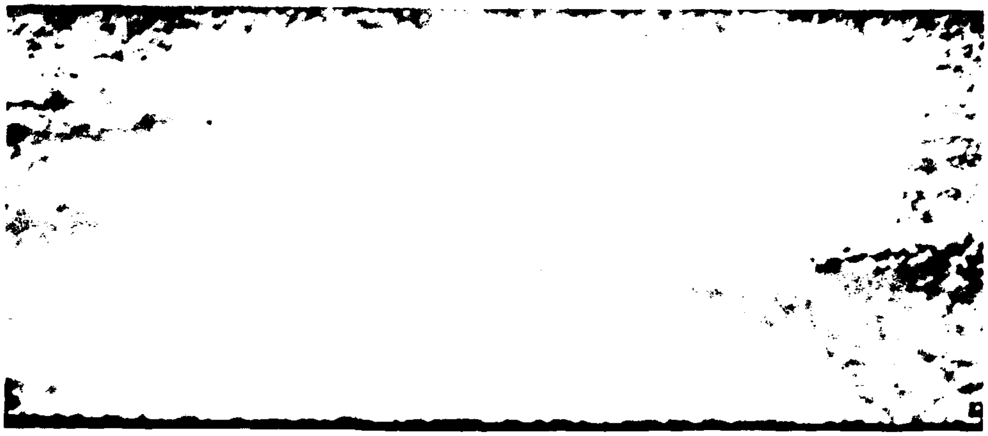


b. Edge Points

Figure 12. Segmentation Using PATS

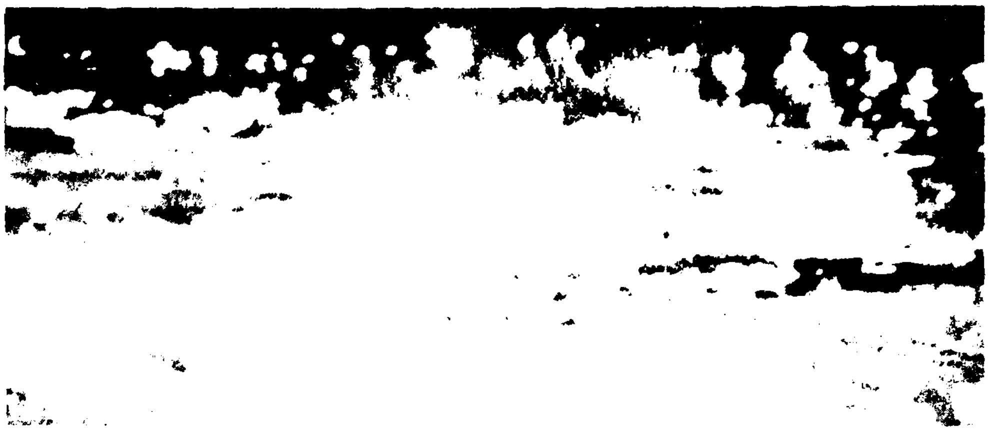


c. Hot Areas

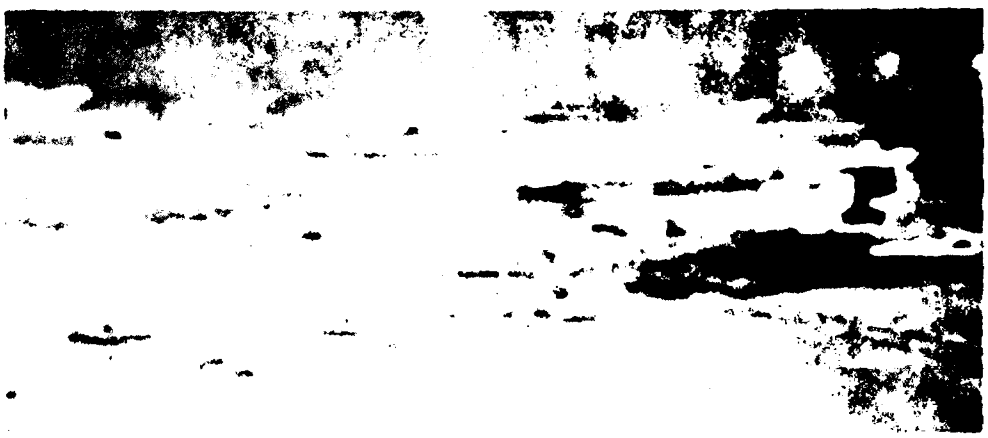


d. Cold Areas

Figure 12. Segmentation Using PATS (continued)

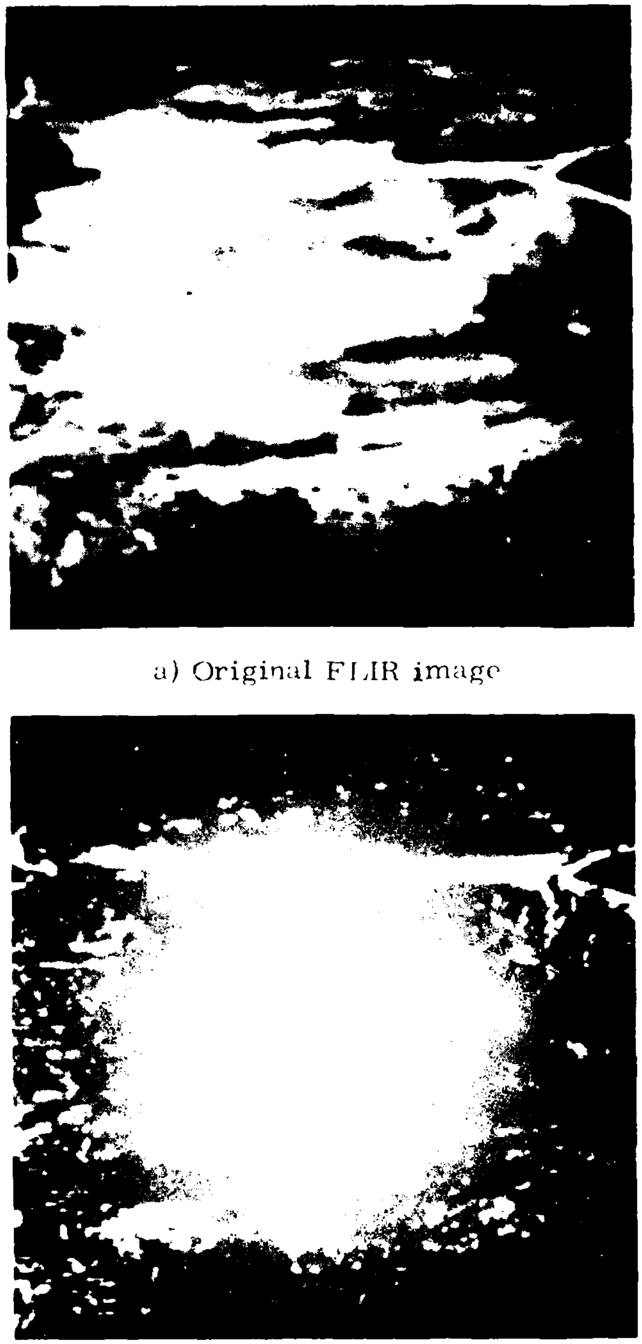


e) Intervals



f) Segmented objects

Figure 12. Segmentation Using PATS (concluded)



a) Original FLIR image

b) Sobel edge image

Figure 13. A Natural Scene and Its Sobel Edge Image

they provide about the matching transformation, their small number, and their relative invariance with respect to changing viewing aspects. Long lines found in an image will be few in number. This means that not many long line matches need be tested before arriving at the correct match, which is important for an algorithm which is to operate in real time. Furthermore, long edges are quite immune to occlusion. Even though some object such as a truck or a building obscures different parts of the edge of a road in two different views of a scene, it will still be possible to determine where the line of the road is in both images from the unoccluded edges. For these reasons, long edges should be classified as distinctive features to be examined early in the execution of the matching algorithm. An overview of the approach used for extracting edge features is shown in Figure 14.

Sobel Edge Operator

The first step in the process of extracting edge features from an image as implemented at Honeywell under the present program is the calculation of the Sobel edge image from the original image. The intensity of the Sobel

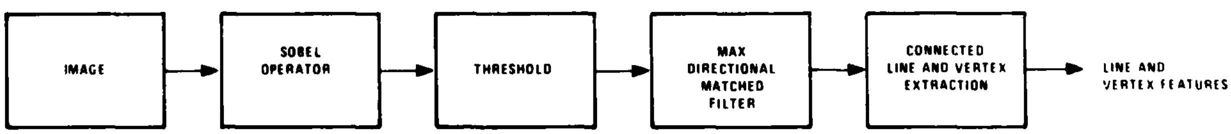


Figure 14. Overview of Edge-Based Feature Extraction

image at pixel (i,j) is given by an equation where f(k,l) is the intensity of the original image at pixel (k,l)

$$
\begin{aligned}
s(i,j) = & \,|f(i+1, j-1) + 2f(i+1, j) + f(i+1, j+1) \\
& -f(i-1, j-1) - 2f(i-1, j) - f(i-1, j+1)| \\
& +|f(i-1, j+1) + 2f(i, j+1) + f(i+1, j+1) \\
& -f(i-1, j-1) - 2f(i, j-1) - f(i+1, j-1)|
\end{aligned}
$$

An algorithm using the parallel computational capabilities of the I^2S Model 70E image computer at Honeywell calculates the Sobel edge image for a full 512 x 512 image in 10 seconds, which is 9 times faster than an algorithm that calculates the Sobel edge operator in the host computer. The parallel algorithm executes in three passes. In the first two passes it determines and saves the signs of the two sums inside the absolute value bars for each pixel, and in the third pass it accumulates the Sobel edge image, using the predetermined signs to know whether to add or subtract each term inside the absolute value bars. The Model 70E and Honeywell's Parallel Image Processing (PIP) chip have similar parallel computation capabilities. PIP's parallelism offers great speed and also compactness for the eventual implementation parts of this project.

The second step in extracting edge features is to threshold the Sobel edge image to obtain a binary edge image which is used as input for the next processing step, maximum directional edge filtering. The Sobel edge images and the thresholded Sobel edge images for the image pair of Figure 9 are shown in Figure 15.

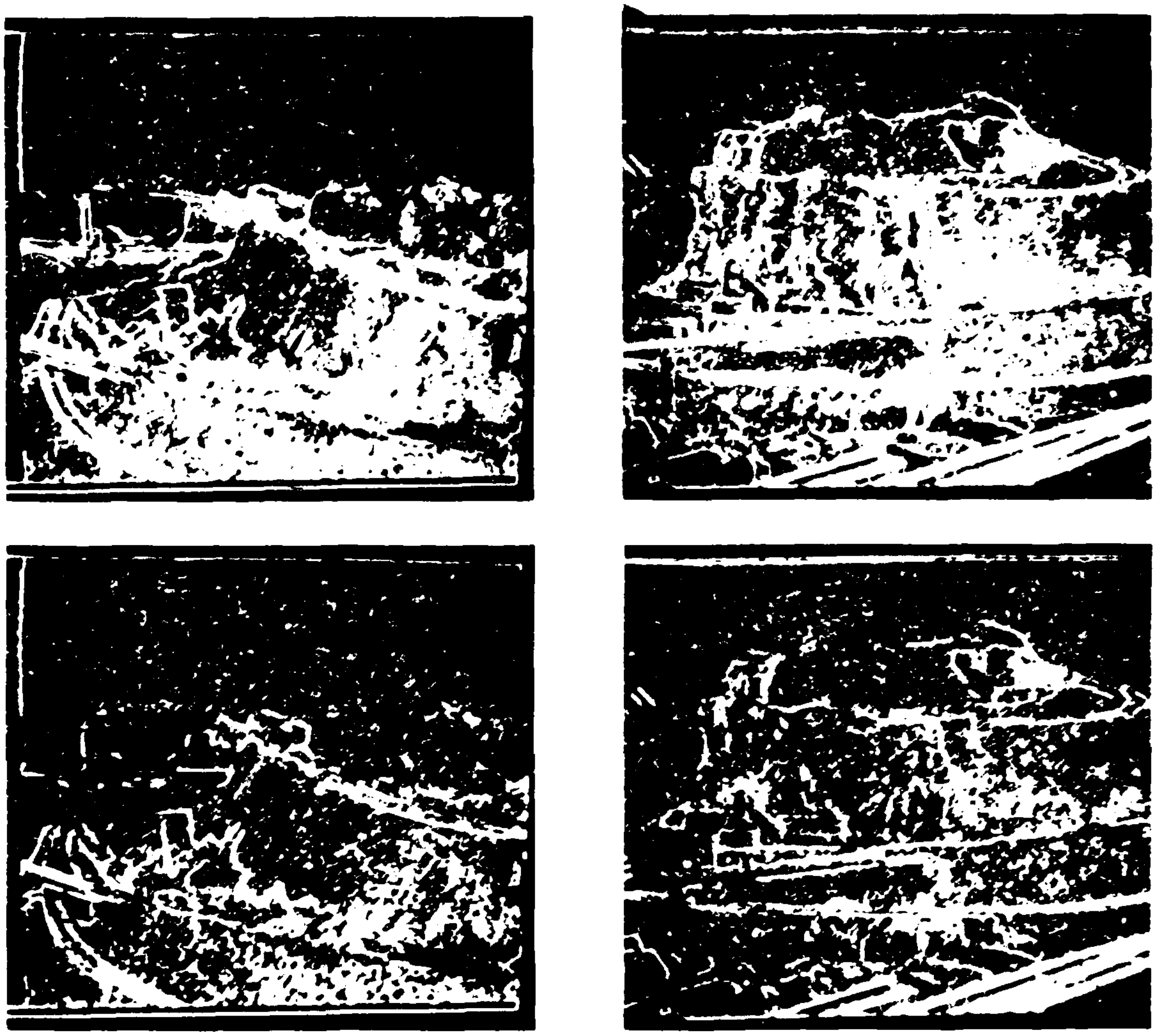


Figure 10. Sobel Edge Images (Top) and Thresholded Sobel Edge Images (Bottom) for the Image Pair of Figure 9

Maximum Directional Edge Filtering

Typically the binary edge image obtained from the Sobel edge image contains points that contribute to both straight lines and undesired clutter points. Maximum directional edge filtering is applied to the binary edge image to eliminate noisy edge points and isolate points that belong to straight lines (or line segments).

An edge filter with length N and orientation angle θ is defined as the filter for which the output at pixel (i, j) is the number of edge points in the binary image in a rectangular region of width 1 and length N centered at (i, j) and oriented at angle θ(Figure 16). Such a filter is essentially a matched filter for a line segment of length N at angle θ. This filter detects points that contribute to straight lines at angle θ by accepting only those points for which the edge filter response was above some threshold. This procedure rejects isolated clutter points and line segments shorter than the threshold.

Maximum directional edge filtering is based on matched-edge filtering. As originally conceived, a maximum directional edge filter computes at each pixel the responses to M edge filters with orientation angles equally spaced through 180 deg. It then outputs both the response and orientation for the filter with maximum response at each pixel. This produces information about the orientation of edges that is used in the next processing step. As with simple matched-edge filtering, one also has the response data and may reject isolated clutter points simply by rejecting points at which the maximum response is below a threshold.

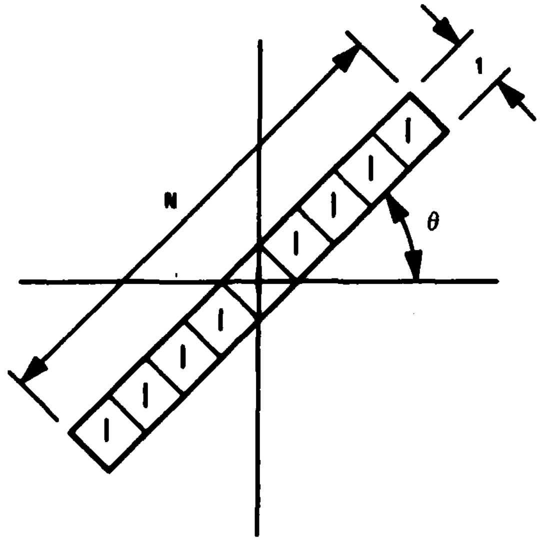


Figure 16. Directional Matched Filter for Edge Enhancement

Results of applying the original maximum directional edge filter to the binary images of Figure 15 are shown in Figures 17 and 18. The edge filter length for the images was 15 and the number of angles was 16. Shown in Figure 17 are only those pixels for which (1) the angle of the maximum response was approximately 11° below horizontal and (2) the response was 8 or greater. This information about the direction of the edges is presented to the edge-linking algorithm, which is the next step in the edge feature extraction process. Shown in Figure 18 are all pixels for which the maximum response was 8 or greater.

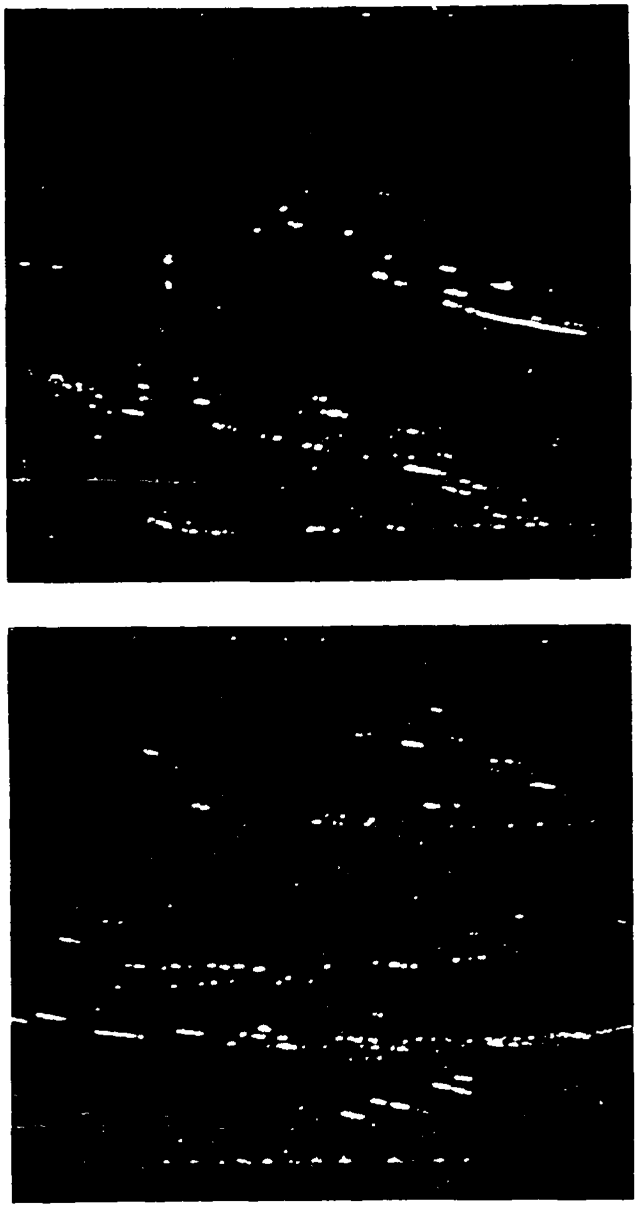


Figure 17. Maximum Directional Filter Output for the Images of Figure 15 Only pixels with response above a threshold and with maximum direction of 10 below horizontal are shown in this figure.

Figure 18. Maximum Direction Filter Output for Images of Figure 15.

All pixels with response above a threshold are shown.

The amount of work that the edge-linking algorithm must do depends on how clean the ouput of the maximum directional edge filter is. A modification of the maximum directional edge filter which incorporates orthogonal suppression filters was proposed to get cleaner output. The output of the orthogonal suppression filter for a given edge filter is the number of binary image edge elements found in two rectangular regions symmetrically placed along a line orthogonal to the center line of the edge filter as shown in Figure 19. The rectangular regions are one pixel wide and their ends are at distances N_1 and N_2 from the center line of the edge filter. The output of the modified maximum directional edge filter is zero if the response for the orthogonal supression filter is greater than or equal to the response for the edge filter. Otherwise, the output is the edge filter response minus the orthogonal filter response. This operation will reduce the output of the modified filter in regions with heavy clutter more successfully than the original filter.

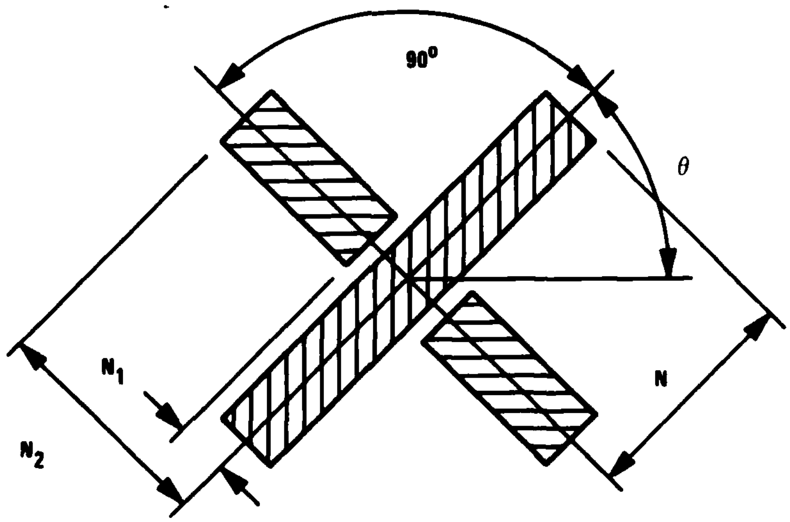


Figure 19. Windows for Edge Filter at Angle θ and its Associated Orthogonal Supression Filter

The original algorithm requires MN additions per pixel where N is the maximum edge-filter response and M is the number of directions or orientation angles. The modified algorithm requires approximately twice as many additions. The algorithms as they are implemented use the parallel computational capabilities of the I^2S Model 70E image computer to achieve efficiency. The binary input image in one Model 70E refresh memory is translated left and right and up and down to align the appropriate pixels of the binary image with those of another refresh memory in which the responses to edge filters and orthogonal supression filters are accumulated. A third refresh memory is used to store the direction and response for the maximum response. The comparisons required for determining the maximum response are implemented in the lookup tables in the Model 70E.

Edge-Linking Algorithm

The edge-linker algorithm acts on the output of the maximum directional edge filter to link pixels into line segments. It considers only those pixels with maximum response above a threshold T1. A section of a thresholded maximum direction image is shown in Figure 20. A zero corresponds to a maximum response which is below threshold T1. Non-zero entries correspond to the direction number of the maximum response if the maximum response was above the threshold.

The edge linker links segments in two passes. In the first pass it reads in one line at a time from top to bottom and links pixels whose maximum directions are closer to vertical than horizontal. In the second pass it reads columns from left to right and links pixels whose maximum directions are closer to horizontal than vertical. The operation of the edge linker will be described for the first pass.

LENGTH=15 ANGLES=16 THRESHOLD= 8

	050	051	052	053	054	055	056	057	058	059	060	061	062	063	064	065	066	067	068	069	070	071	072	073	074	075	076
300	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	16	0
301	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
302	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
303	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
304	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0
305	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
306	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
307	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
308	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
309	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
310	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
311	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
312	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
313	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0
314	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
315	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0
316	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
317	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
318	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0
319	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0
320	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
321	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
322	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0
323	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	14	0	0	0	0	0	0	0	0	0	0	0
324	0	0	0	0	0	0	0	0	0	0	0	0	0	13	15	0	0	0	0	0	0	0	0	0	0	0	0
325	0	0	0	0	0	0	0	0	0	0	0	0	14	15	0	0	0	0	0	0	0	0	0	0	0	0	0
326	0	0	0	0	0	0	0	0	0	0	0	13	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
327	0	0	0	0	0	0	0	0	0	0	13	14	14	16	0	0	0	0	0	0	0	0	0	0	0	0	0
328	0	0	0	0	0	0	0	1	7	15	13	6	0	16	1	0	0	0	0	0	0	0	0	0	0	0	0
329	0	0	0	0	0	0	0	1	16	8	7	7	7	6	1	0	0	0	0	0	0	0	0	0	0	0	0
330	0	0	0	0	0	0	2	1	16	15	14	6	7	7	7	0	0	0	0	0	0	0	0	0	0	0	0
331	0	0	0	0	0	0	1	1	16	14	0	5	6	6	7	7	7	0	0	0	0	0	0	0	0	0	0
332	0	0	0	0	0	0	1	16	16	0	8	7	7	0	6	6	6	7	7	7	0	0	0	0	0	0	0
333	0	0	0	0	0	0	1	16	15	0	0	5	0	7	7	7	0	6	6	6	7	7	7	9	9	9	9
334	0	0	0	0	0	0	1	16	15	0	0	0	11	0	0	0	7	7	7	6	6	6	0	7	7	7	8
335	0	0	0	0	0	0	1	16	0	0	0	0	0	0	0	0	0	6	0	7	7	7	6	6	9	9	7
336	0	0	0	0	0	0	16	16	0	0	0	0	0	0	0	0	0	7	0	0	0	0	7	7	9	9	6
337	0	0	0	0	0	0	16	1	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	8	7	7
338	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	7	0	0	6
339	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0
340	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
341	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
342	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
343	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
344	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
345	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
346	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
347	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
348	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
349	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
350	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
351	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
352	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
353	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0

Figure 20. A Section of a Thresholded Maximum Directional Edge Filter Output. N = 15, M = 16, and T1 = 8.

The data structure on which the algorithm is based is an active segment list. Pointers at the head of the list make it possible to find active segments with a given direction number without searching through active segments with different direction numbers. The data retained in the active segment list are

1. the line in which the segment was started,
2. the center of the segment in the starting line,
3. the last line in which a pixel was added to the segment,
4. the center of the lines in that last line, and
5. data to determine the regression line for the segment.

The edge-linking algorithm for the first pass is as follows (Figure 21):

1. Initialize the active segment list as empty.
2. Repeat the following for each line from the top of the image to the bottom :

   a. Read in the next line to be processed.

   b. For each pixel in the line above threshold T1, find the active segment with the same direction number which has its center line nearest the new pixel. If the distance to the center line is less than a threshold T2, add the new pixel and all pixels within T2 of the new pixel with the same direction number to the active segment. Of course, the last line in which a pixel was added to the segment and the centroid in that line must be updated. T2 is the line halfwidth threshold.

c. "Drop" those active segments that have not had pixels added for more than T3 lines. T3 is the threshold for the width of a gap in a line that the edge linker will bridge.

d. Create new active segments for those pixels above threshold T1 that were not added to any active segment in step 2b. Group pixels within 2*T2 of each other and with the same direction number in the same new active segment.

e. After the last line is processed, "drop" all active segments.

Point A is the center of the active segment in the last line in which pixels were added to the active segment. Line AB extends from A in the direction of the segment. The segment may be extended if pixels with response above threshold T1 and direction number equal to that of the segment are found in the search region.

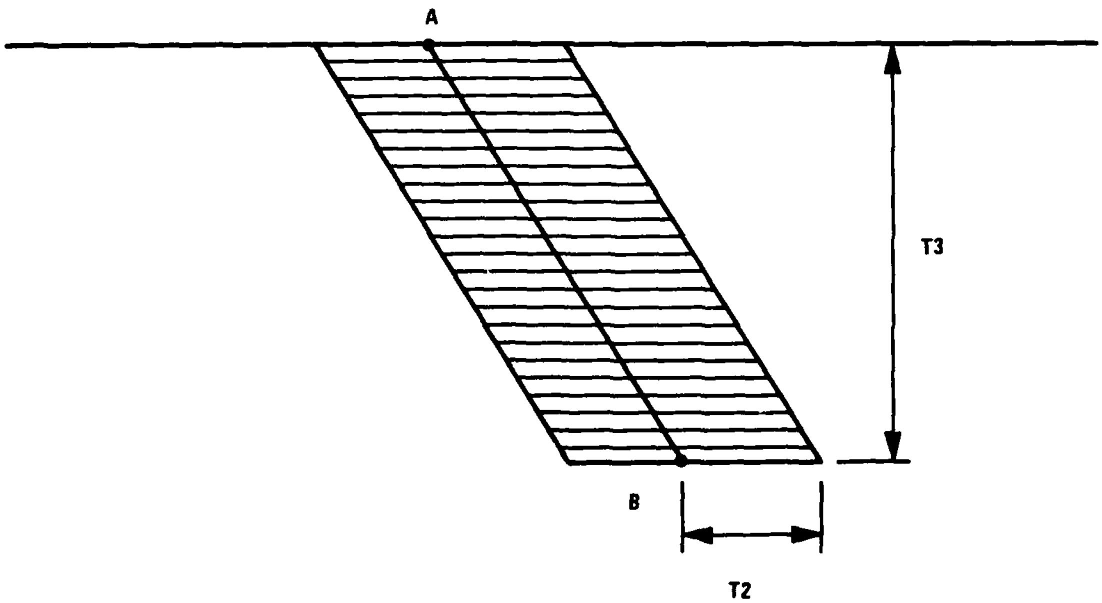


Figure 21. Search Region for Extending an Active Segment

When an active segment is "dropped," two things happen. First, if the dropped segment is longer than a threshold T4, the data in the active segment list describing it is written onto an output file that will be used by the segment merger algorithm. Second, its space in the active segment list is freed up for new active segments.

Figure 22 shows the result of the edge-linking algorithm on the output of the thresholded directional edge filter in Figure 18. T1 is typically about half the maximum possible response for the maximum directional edge filter. This choice of T1 seems reasonable because smaller T1 will tend to lengthen lines and larger T1 will tend to shorten lines. Also, this choice of T1 was found to be best experimentally. The choice of T2, the line halfwidth threshold, is governed by the line thickness in the maximum directional edge filter image. In Figure 22 that thickness is 4 pixels. 2*T2 should be slightly larger. T4 governs the number of segments that the segment merger must consider. T4 has been set at 7 pixels.

Line Segment Merger

As explained above, long edges are particularly useful edge features for matching. Long edges may be expected to be partly occluded so that they appear as multiple segments along the same line. The gaps between the segments will generally be much larger than the gap that the edge linker may be allowed to bridge. This means that another processing step is required after edge linking--segment merging. Also, the edge linker in its initial implementation tends to generate close parallel line segments that really ought to be merged because it processes the pixels with

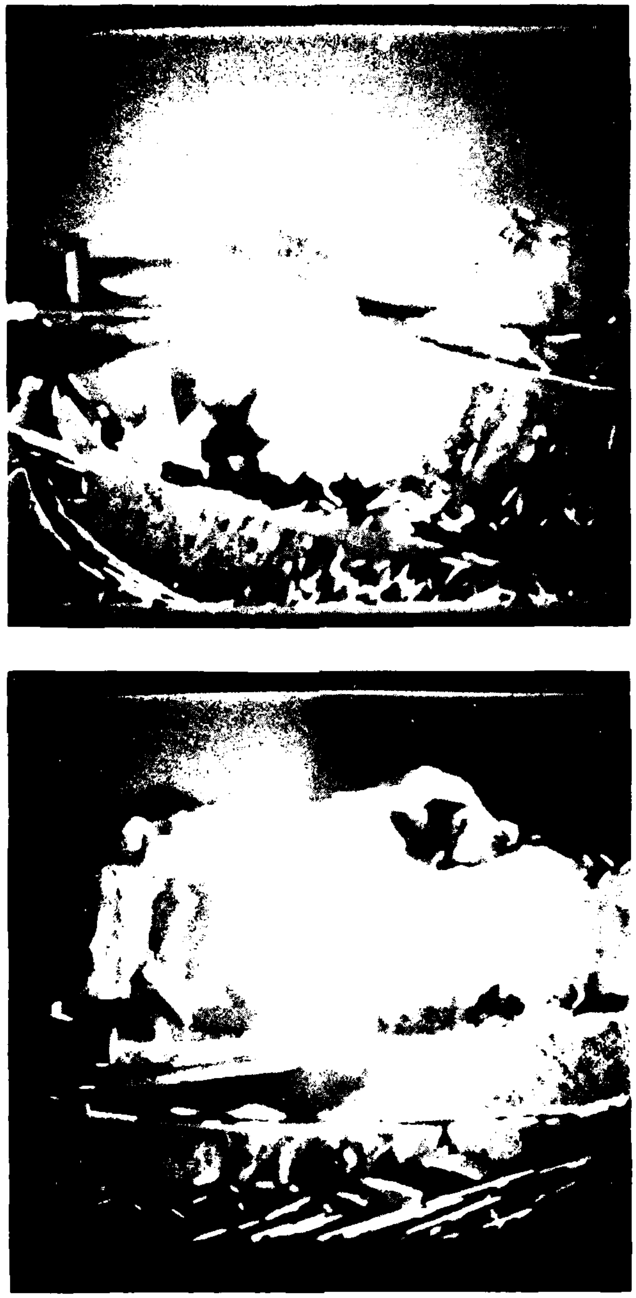


Figure 22. Edge Linker Output for the Images of Figure 18.

different maximum direction separately without merging. Segment merging can be used to eliminate redundancies in the segment list generated by the edge linker as well as to detect partially occluded long lines.

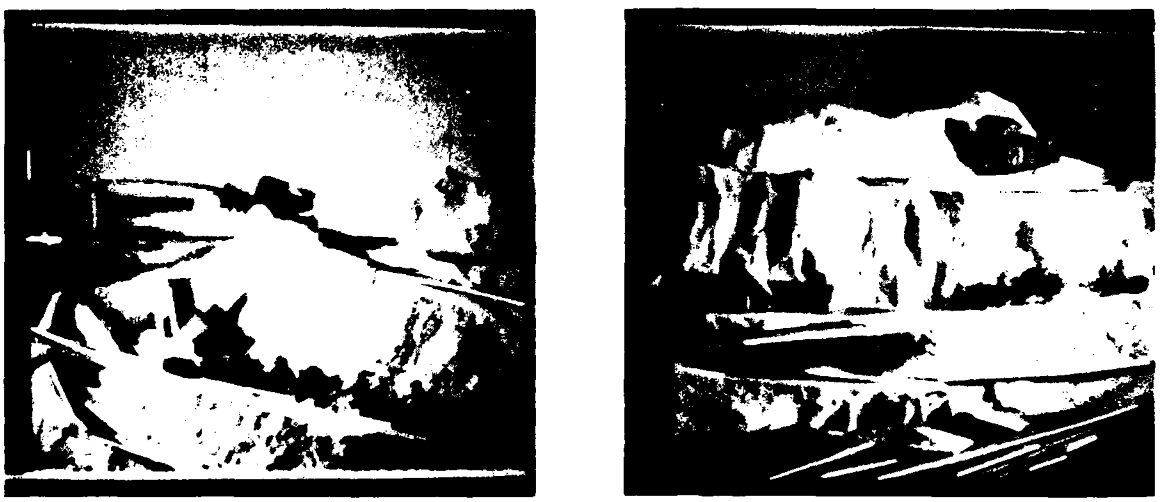
The segment merger starts with a segment list, either one generated by the edge linker or one generated by a previous execution of the segment linker. It examines pairs of segments, and when it finds a pair that meet criteria for merging, it deletes the two segments from its segment list and adds a segment that is obtained by merging them. It then continues its search for segments that may be merged. Criteria for merging are: the difference between the angular displacements of the segments with respect to horizontal, the gap between the merged segments, the width of the merged segments and the ratios of the gap, and the width-to-length ratio of the merged segment. The edge linker tends to generate a list of segments which has segments that should be merged close to each other in the list. Therefore, the segment merger initially examines only those pairs of segments that are close to each other to reduce the number of comparisons that must be made. After the first stage of merging is complete and the segment list has been collapsed, the segment merger operates on the entire list.

Two applications of the segment merger are made to each segment list generated by the edge linker. The first application is used to eliminate redundant segments without merging beyond the segment level. The second application is used to obtain long edges. The two applications require different merging criteria. In Figure 23 the results of applying the segment merger to the segment list generated by the edge linker for the maximum direction edge filter images of Figure 22 are shown. The segments remaining after

(a)

223 segments extracted from 532 generated by edge linker

103 segments extracted from 344 generated by edge linker



(b)

20 long lines extracted

19 long lines extracted

Figure 23. Line Segment Merger Output

The first pass outputs segments which are shown in the top pictures. The second pass outputs long lines which are shown in the lower pictures.

segment merging to eliminate redundancies in the list are shown in Figure 23a. The long segments extracted from the reduced segment list(s) are shown in Figure 23b.

SECTION V

PLANS FOR THE NEXT REPORTING PERIOD

In this section we summarize our plans for the next reporting period. The main thrust of our activity in this reporting period will be in:

- Object-matching criteria
- Configuration-matching criteria
- Branch and bound search algorithm

OBJECT-MATCHING CRITERIA

In the next reporting period we will complete the evaluation of the object segment feature extraction alternative--Prototype Similarity and PATS segmentation. Because Prototype Similarity gives a large number of multiple segments per object, we will examine techniques for merging object segments before using them in the symbolic description of the image. The edge-based feature extractor will also be evaluated with FLIR images to determine its utility. We will develop criteria for comparing objects based on the descriptors. The major issue to be addressed will be the invariance of the features to sensor geometry and sensor characteristics. For example, the shape of the objects extracted will depend upon the scene aspect and perspective. It is expected that descriptors for FLIR images will prove to be far more tractable than those for visual images.

CONFIGURATION-MATCHING CRITERIA

We will examine techniques for evaluating the goodness of a match between two sets of object configurations which satisfy topological and geometric consistency tests. Because of the nature of the autonomous acquisition scenario, we will relax all six degrees of freedom in testing for consistency of configurations. An analytic task is the derivation of goodness of match criteria which includes matching line features between the two images as well as point features.

BRANCH AND BOUND SEARCH ALGORITHM DEVELOPMENT

We will specify the enumeration procedure for the branch and bound search algorithm. Software development of the branch and bound matching program will begin.

We will continue in our quest for adequate FLIR imagery for the pattern-matching simulation task. In the absence of imagery representing the entire range of sensor geometry and characteristics, it may be necessary to simulate changes in viewing aspects and perspective from existing FLIR scenes. However, the emphasis will be on evaluating the algorithms on real imagery which is characteristic of the autonomous acquisition scenario.